
POPULAR
Computing
WEEKLY
6 May 1982 Vol 1 No 3
30p
Hell Drivers
on Vic-20

MORE FUN WITH YOUR ZX81!

16K RAM PACK

PUSH-ON PLUG-COMPATIBLE MEMORY

The addition of our fully-compatible, assembled, tested and guaranteed 16K Ram Pack means **more memory for better games and programs**

ORDER YOURS TODAY FROM

CAPS LTD Dept E, 28 The Spain, Petersfield, Hants GU32 3LA

Please send me my Byg Byte 16K Ram Pack. I enclose cheque/postal order for £34.95 (includes VAT, Postage & Packing)

Name

Address

Dept E

Allow 28 days for delivery

Fully **£34.95** inclusive

From West London's mail order specialists:

ONLY £240
incl. cassette deck
+ FREE GAME
+ FREE DELIVERY

VIC 20

SPECIAL OFFER

Order the superb VIC 20 and cassette deck at £240 (incl VAT!) and get a FREE game ("Breakout" or "Stunt Cycle") worth £7.99 – and delivery is free too!
We stock all VIC 20 software, peripherals, paddles, etc – including the full Stack and Arfon range.

The best microcomputers from leading makers (Apple, Commodore, etc) plus peripherals, printers, software, games, books – and all at low prices! Order today from the mail order specialists – send coupon with cheque (or phone your Barclaycard no). Delivery in 28 days or money-back option.

NAME
ADDRESS

TWICKENHAM COMPUTER CENTRE LTD
72 Heath Rd Twickenham Middx TW1 4BW (01 892 7896/01 891 1612)

From West London's mail order specialists:

Latest special offers:

48K Apple	£768
12" Green monitor	£115
12" Amber Phillips monitor	£135
Epsom printer	£399
Tantel adaptor	£230

ALL PRICES INCLUDE VAT!

TWICKENHAM COMPUTER CENTRE LTD
72 Heath Rd Twickenham Middx TW1 4BW (01 892 7896/01 891 1612)

POPULAR Computing WEEKLY

The Team

Editor
Duncan Scot

Production consultants
Myles Hewitt
Eric Robbie

Editorial Secretary
Fiona McCormick

Advertisement Manager
David Lake (01-839 2846)

Advertisement Executive
Peter Chandler (01-839 1881)

Software Editor
Peter Gerrard (01-839 1655)

Publishing Director
Nick Hampshire

Popular Computing Weekly
Hobhouse Court
19 Whitcomb Street,
London WC2 7HF
01-839 6835 and 01-839 1855

Published by Sunshine Publications Ltd, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF

Typesetting origination and printing by Chesham Press Chesham Bucks

Distributed by S M Distribution London SW9 01-274 8611 Telex 261643

Subscriptions
You can have *Popular Computing Weekly* sent to your home the subscription rate is £19.95 per year for addresses in the UK

How to submit articles
Articles which are submitted for publication should not be more than 1000 words long

All submissions should be typed and a double space should be left between each line

Programs should whenever possible be computer printed

At present we cannot guarantee to return every submitted article, so please keep a copy

Accuracy
Popular Computing Weekly cannot accept any responsibility for any errors in programs we publish although we will always try our best to make sure programs work

This Week



Cover illustration by Ian Craig

Editorial

A couple of years ago the BBC had a good idea. The corporation decided that it ought to do something about the personal computer revolution

There was to be a series of computer literacy programmes and a specially designed BBC Microcomputer to accompany them

So what went wrong?

The BBC took too long to decide on the design and manufacture of the computer and allowed too little time for problems to be ironed out

This meant the computers were not sent out in time for the broadcast programmes

The BBC also seriously underestimated the demand for the computer. Their guess was 12,000 units. In fact they will probably sell well over 50,000 this year

The whole fiasco says far more about the BBC than it does about the personal computer market, booming year after year as eager entrepreneurs seize hold of a slice of the market

Would that British industry, broadcasting, and Government were so unfettered by caution and timidity

Next Week



Want to learn your innermost secrets? Play Mind Reader and you'll discover things you never even knew you knew.

Vic gives specs for two

At the recent Hanover Fair Commodore were able to announce quite detailed specifications on their new range of microcomputers including the Ultimax (now renamed the Vic 10) and the Vic-30

The Vic-10 has an anticipated selling price of £100 and an expected launch date of September this year Onboard RAM is just 2K with no Basic to play with

It is the sound and graphics capability which will attract most attention The text screen area is 40 x 25 (much better than the original Vic) with a high resolution mode of 320 x 200 addressable pixels

With the new 6566 chip creating true three-dimensional graphics becomes extremely simple

Sound is controlled by the SID chip This allows for three voices each with a nine octave range A programmable filter is also inbuilt for each voice (Mike Oldfield, eat your heart out!)

Contrary to earlier rumours, the Vic-10 can interface to other peripherals. Most advantageous will be the cassette port

The Vic-30 is essentially a 'big brother' to the other machine, having a Vic-20 style keyboard, but retaining the 40 x 25 character display on the screen

You have 16K of RAM on board, and also a full Basic operating system There is a serial interface port, memory expansion port, and a cassette interface port all built in

Delivery is scheduled to commence in January of 1983 and the expected retail price will be £250



Martin Hayman

Dynamic duo go Dutch in Prestel competition

Marco van Gent (*left*) and Ari Schot, both 16, have every reason to look pleased with themselves

The two young Dutchmen from Leiden, Holland have just carried away the runners-up prizes in the Prestel competition to build a telesoftware adaptor for the Sinclair ZX81

Marco and Ari arrived in London on the Saturday before the contest with only a £70 girocheque between them — which they couldn't cash Prestel's Tony Sweet took pity on them and put them up until the post offices opened on Monday morning

Marco and Ari will now be co-operating with co winners of the contest Lion Viewdata in the production of the telesoftware adaptor Lion will develop the hardware and the Dutchmen will do the software Eventually Lion hope to offer an adaptor for other micros as well as the ZX81

The first program programme begins

Radio WM has started a new regular radio programme on home computing The first of the series was broadcast on Saturday 24 April by the BBC local radio station, which covers the whole of the West Midlands area

The programmes, brainchild of producer Tim Manning, are broadcast between 8 45 am and 9 00 am during the *Great! It's Saturday* morning slot

Tim Manning believes it is the first time a radio station has offered a regular programme for home computer users He plans to start at a very simple level, giving advice about how to choose a computer

Future plans include finding a way of introducing people to programming and, at some stage, broadcasting Basic programs over the air

They'll fit a ZX81 printer to anything

Own a Vic-20 but cannot afford a printer? Now you can buy a Sinclair ZX printer for £59 95, and link it up to your Vic

The interface is being sold by Microtanic Software and there are versions for the BBC Atom, Microtan and Aim-65 computers

The Printerface costs £29 95 inclusive

The address is on page 22

Sinclair Spectrum stuns BBC

The ZX Spectrum has arrived Clive Sinclair announced the release of his new computer on the opening morning of the Computer Fair at Earls Court on Friday 23 April (Did Sinclair know that it was St George's Day?)

The basic model comes with 16K of memory, full colour and sound, and costs £125 A full report on the announcements and the world's first hands-on review of the ZX Spectrum starts on page 10

The reaction to the news of Sinclair's new computer was immediate There is now strong speculation that Commodore will have to reduce the price of the Vic 20 to well under £100 with corresponding reductions on the prices of the peripherals

Acorn has also dismissed Sinclair's suggestion that the ZX Spectrum will undermine the BBC Microcomputer Speaking privately to *Popular Computing Weekly* Acorn director Herman Hauser said that a new computer would be launched in the third quarter of this year — sometime between July and September

The new Acorn computer will probably be called the Electron and cost between £120 and £150 There will be a 32K ROM and 32K of RAM and in effect the machine will be a miniaturised BBC Microcomputer with higher resolution graphics than those offered by the Spectrum

● GOTO page 10!



The PC 1500 is a new handheld computer from Sharp and is an enhanced and extended version of the PC-1211 The new computer has an optional 4-colour graphic printer/cassette interface, and is programmed in Basic There is only 3 5K of internal memory, but this can be increased to 7 5K with a 4K module It costs £169 95 and the cassette/printer interface is £149 95

Club Reports

Is your club involved in any special projects? Use this page to tell the world about it.

Thousands roll up for all the fun of the fair

***Mike Todd** reports on the London computer clubs' annual exhibition*

More than 5,000 computer enthusiasts turned up at the Polytechnic of North London for the Association of London Computer Clubs' third Computer Fair

Two years ago, computer clubs in the London area decided they wanted their own exhibition catering for their own needs At that time most exhibitions were oriented towards the business user

They also felt a need to start to co-ordinate the activities of the London clubs and to assist those wishing to start clubs in their area

To that end, they got together and formed the Association of London Computer Clubs, and the group now numbers a dozen associated clubs, all of which hold regular meetings in London and neighbouring counties

If you live in the London area and have a club which is not part of the ALCC, or would like to form a new club, get in touch with the ALCC secretary Len Stuart at 89 Mayfair Ave, Worcester Park, Surrey KT4 7SJ If you've got Prestel, you'll find information on the activities of most of the ALCC groups on page 35748

Inter-computer communications

With data and inter-computer communication at the forefront of today's micro development, it is appropriate that the Amateur Computer Club, or ACC — not to be confused with the ALCC — should mount a seminar during the exhibition on the computer and the telephone

About 40 people attended, and they were taken from the concept of Information Technology, through the interfaces and modems required for practical applications They were also treated to a demonstration of British Telecom's Prestel service

The seminar ended with a discussion of the use of the personal computer for 'bulletin board' applications, and of computerised mailboxes which will become widespread over the next year or so With cheap modems becoming available, most microcomputer owners should be able to take advantage of this aspect of information technology

A wide range of interests were catered for at the London clubs computer fair

The Independent Commodore Product Users' Group (ICPUG) held a seminar of a slightly different kind In one of the rooms (a bit of a slog from the main exhibition hall) they set up several Vics showing a range of games During the course of the day, there were two programmers' clinics and a demonstration of Superscript which has to be the lowest cost, quality word processor available

The groups that make up the ALCC were relegated to the bowels of the Poly where they showed off their software and hardware projects, while upstairs in the main theatre 40 exhibitors showed their wares These included national computer groups such as the ICPUG and the ACC, among them were the Sinclair User Group showing what they had to offer

A number of dealers displayed their wares, and reported brisk sales of a wide range of devices including disks disk drives, RAM chips and printers With some of the discounts being offered, it was hardly surprising that their stands were busy

By far the greatest crowds seemed to be the large numbers of small boys sitting at Vics, Ataris and Apples and playing just about every game imaginable! The BBC micro which certainly attracted a great deal of attention, but the greatest 'buzz' running through the fair must have been the rumours about the new Sinclair ZX82 — the Spectrum

Mice at the speed of sound

The heats of the Micro-Mouse competition were held in a crowded side room For the uninitiated, this is a competition to find the robot mouse which can find its way fastest through a maze

This task is not as easy as you might imagine, and there were only two competitors at the fair, although one of these, Alan Dibley, had three Mice running — Thezarus, Son of Thezarus, and T3

With a ZX80 at the heart and a selection of model aircraft motors they unfortunately failed to beat Stirling Mouse entered by Nick Smith which was fastest in both attempts at the maze with a best time of 1'46"

Write to Club Reports *Popular Computing Weekly* Hobhouse Court, 19 Whitcomb Street, London WC2 7HF with details of successes you have had with your club, with ideas for helping clubs along and with any news of special meetings We look forward to hearing from you

Hell Driver

Race against time and trouble in this high-speed action game

by Dave Middleton

SET THE controls for the heart of the road, as you assume command of a skimmer racing around the track You have to get round the track in as short a time as possible, but you must be careful about trying to go too fast

Your vehicle is equipped with four gears, and the scanner display keeps you constantly informed of which gear you are in, as well as showing you the speed at which you're travelling and the number of revs you're doing

Keep a wary eye on your speed, and don't try and move away too fast. It is possible to over-rev your machine, and your engine can only handle so much You will be warned if you are reaching danger point, and you'll have to very quickly take appropriate action to get out of trouble in time

The track abounds with other obstacles to get past, so travelling at the right speed at the right time is vitally important You have to keep a constant look out to avoid crashing

The Game

There are six keys you must use almost all the time For accelerating and braking, the keys A and B respectively do the trick Changing gear is done by using the four function keys on the right of the Vic

An examination of the program listing with this article will reveal how these remarkably undocumented keys can actually be used within a program Lines 410 to 480 will do the trick for you

The program works on the standard Vic, without any memory expansion required, and makes reasonable use of the Vic's built-in sound and colour capabilities

Apart from the keys mentioned earlier only one is required throughout the game and that is simply pressing S to start the game rolling

Then, quite simply, you have to get round the track in the shortest possible time, and try and avoid all the hazards mentioned earlier

On the first few runs through that is quite a challenge, and you'll probably turn the sound down on your television as you begin to despair of ever completing the course

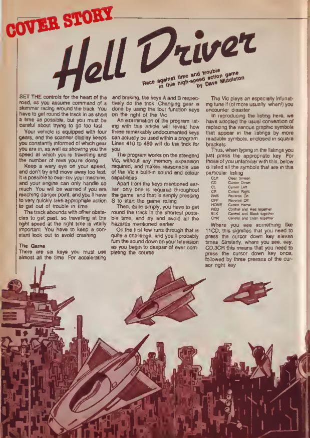
The Vic plays an especially infuriating tune if (of more usually when!) you encounter disaster

In reproducing the listing here, we have adopted the usual convention of replacing the various graphic symbols that appear in the listings by more readable symbols, enclosed in square brackets

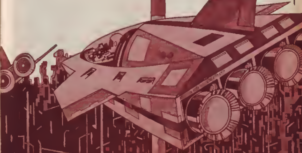
Thus, when typing in the listings you just press the appropriate key For those of you unfamiliar with this, below is listed all the symbols that are in this particular listing

CLR	Clear Screen
CD	Cursor Down
CL	Cursor Left
CR	Cursor Right
RVS	Reverse On
OFF	Reverse Off
HOME	Cursor Home
RED	Control and Red together
BLK	Control and Black together
CYN	Control and Cyan together

Where you see something like 11CD, this signifies that you need to press the cursor down key eleven times Similarly, where you see, say, CD,3CR this means that you need to press the cursor down key once, followed by three presses of the cursor right key



```
90 DIM H1 14  B1 14  C1 14  [illegible] B5(10) M6 14
90 [illegible] C=38400+[illegible] [illegible] P=0, S=0
100 PRINT [illegible]
110 FOR Z=1 TO 14  READ A B   A1 Z =A  B1 Z =B
    C1 Z =C
120 FOR D=1 TO A  POKE 36 [illegible] +B  POKE C [illegible] C0=C0+B
    NEXT D
130 [illegible] C0=C0+C  NEXT Z
140 DATA [illegible] 3 [illegible] -2  1 [illegible] 2 -1  4
    [illegible] -1 [illegible] -1 [illegible] 3 [illegible] 2
150 DATA [illegible] 1 1 2 -1 -22 [illegible] 1 -22 4 [illegible] 22
    1 [illegible] 3 -2  2 4  21 1
160 FOR Z=1 TO 5  READ C5 Z   NEXT
170 DATA [illegible] 4 1 9 1 5 1,2 1
180 FOR Z=0 TO 9  READ B5 Z   NEXT
190 DATA 0 [illegible] 15 20 25 30 40 [illegible] 36 [illegible]
200 FOR Z=1 TO 14  READ M6 Z   NEXT
210 DATA [illegible] 25 40 [illegible] 90 80
    4[illegible] 200
220 POKE [illegible] 15
230 D=10
250 POKE [illegible]
300 [illegible] C=38400+2[illegible] [illegible] T1 B=A
305 FOR [illegible] STEP 10  POKE [illegible] 12[illegible]
    FOR [illegible] -1 TO 1[illegible] NEXT  NEXT  GOSUB [illegible]
310 A=[illegible]
320 FOR Z=1 TO 14
330 A2=A1 Z   B2=B1 Z   C2=C1 Z   M[illegible]=M6 Z
340 D=15  GOSUB [illegible]
350 IF [illegible] 14 THEN PRINT MAX SPEED AT BEND  M
360 FOR DL=1 TO A2
370 D=16 GOSUB [illegible]
380 PRINT [illegible] [CL] [illegible]MPH
    INT [illegible]
410 GET H$
420 IF H$=[illegible] AND HH=0 GOTO 500
430 IF H$=[illegible] AND HH=1 THEN HH=0  GOTO 490
435 H=ASC H$
440 IF H=133 AND [illegible]=2 THEN [illegible]-1
445 IF H=134 AND G=1 OR L=3  THEN [illegible]
446 IF H=135 AND [illegible] OR L=4  THEN G=3
448 IF H=136 AND G=3 THEN G 4
460 IF H$= [illegible] THEN GOSUB [illegible]
470 IF [illegible] THEN GOSUB [illegible]
480 HH=1  GOTO 410
490 GOSUB [illegible]
500 IF [illegible] THEN 540
510 IF R 2000 THEN 580
520 IF R 1[illegible] THEN 560
530 IF R 14000 THEN 610
540 OV=OV+1  IF OV>2 THEN [illegible]
550 D=1[illegible] GOSUB 890  PRINT"[RVS]WARNING[OFF] YOU HAVE [illegible]
    ER-REVVED [illegible]
555 FOR [illegible] -1 TO [illegible] FOR[illegible]=1 TO 10  POKE[illegible] 150
    NEXT  POKE36876 200  NEXT
556 POKE[illegible] 0
560 NL=1  IF [illegible] THEN NL=INT  200-[illegible]
570 GOSUB [illegible]  D=10  GOSUB [illegible]  GOTO [illegible]
580 D=1[illegible] GOSUB 890  PRINT"[RVS]YOU HAVE  STALLED
    [illegible]
590 D=1 [illegible] GOSUB 890  PRINT"[RVS]YOU HAVE SPUN[illegible]ED TO A ST
    OP"
600 B=0  D=1 [illegible] A=3  GOSUB 860  GOTO 560
610 D=1[illegible] GOSUB 890  PRINT"[CLR 11CD RED]YOU HAVE BL[illegible]N
    UP[BLU]"
615 POKE368[illegible] 1[illegible] FOR[illegible]=1 TO 150 POKE[illegible]4 128+RND 1[illegible]
    [illegible] NEXT
616 GOTO [illegible]
[illegible] NEXT DL
640 [illegible]
650 IF [illegible] THEN 690
660 IF [illegible] THEN [illegible]
665 D=1[illegible] GOSUB [illegible]  PRINT"[CLR 11CD [illegible] RED] [illegible]OU HAVE
    [illegible]RASHED![BLU]"
[illegible] POKE[illegible] 128  FOR[illegible]=1 TO 100 POKE[illegible] 1[illegible]+RND [illegible]
    1[illegible] POKE[illegible]4 128+RND [illegible]120 NEXT
[illegible] POKE[illegible] 0  POKE[illegible]876 0  POKE[illegible]4 0  POKE[illegible]
    0  END
[illegible] D=1[illegible] GOSUB [illegible]  PRINT [illegible] HAVE [illegible]KIDDED[BLU]"
[illegible] FOR [illegible] 150 TO [illegible] STEP [illegible] POKE[illegible]
    [illegible] NEXT
[illegible]
[illegible] NEXT Z
[illegible] TS= TI [illegible] TM=INT [illegible]
[illegible] TS=INT T[illegible] TM[illegible]
[illegible] D=1  GOSUB [illegible]
[illegible] POKE[illegible]4 [illegible]
[illegible] PRINT [illegible] TIME WAS [illegible] PRINTTH MO[illegible] T [illegible]
[illegible] END
[illegible] FOR L=0 TO [illegible] L1=[illegible] NL  NEXT
[illegible] RETURN
[illegible]
[illegible] B=B+[illegible]  A=0
[illegible] IF B [illegible] THEN B=[illegible]
[illegible] RETURN
[illegible] H=A+[illegible] B=B  IF A [illegible] THEN [illegible]
[illegible] RETURN
[illegible]
[illegible]
880 POKE[illegible]4 128+[illegible] 150
[illegible] PRINT [HOME]
900 IF D 16 THEN D=D-  NL=1
910 FOR DI=1 TO D  PRINT  NEXT
[illegible] IF NL=0 GOTO [illegible]
[illegible] NL=0  PRINT"
[illegible] RETURN
[illegible] END
1000 REM INSTRUCTION
1010 PRINT [CLR]" [illegible] 4
1020 PRINT YOUR [illegible] IS VERY
1030 PRINT"EASY TO CONTROL
1040 PRINT [CD]GEAR  ARE SELECTED [illegible]
1050 PRINT"PRESSING THE FUNCTION
1060 PRINT KEYS -
[illegible] PRINT"[CD] [illegible] SECOND
1080 PRINT [illegible] THIRD [illegible]FOURTH
[illegible] PRINT [CD]TO A[illegible]ELERATE PRE[illegible]
[illegible] PRINT [CD]T[illegible] BRAKE [illegible]
[illegible] PRINT [CD] [illegible] TART
[illegible] GET H$  IF H$ [illegible] TO 11[illegible]
    PRINT [CLR] CHR$ 14
1140 RETURN
```



EXCLUSIVE!

Review

Sinclair strikes back at the BBC

Nick Hampshire writes the first review of the new £125 ZX Spectrum



At a press conference at the Churchill Hotel on Friday April 23, Clive Sinclair revealed to the waiting world his new microcomputer, the ZX Spectrum.

In comparing the Spectrum to another currently popular micro, the BBC micro model A, Clive Sinclair said: "It is obvious at a glance that the design of the Spectrum is more elegant. What may not be so obvious is that it also provides more power.

"The ZX Spectrum has more usable RAM, and higher maximum RAM. It offers twice as many colours on the screen at any one time, plus a colour brightness control. It also offers user definable graphics.

"It has a data transfer rate 25 per cent faster, supported by a VERIFY facility. And it employs a dialect of BASIC (Sinclair BASIC) already in use in over 400,000 computers worldwide. We believe the BBC makes the best TV programmes — and that Sinclair makes the world's best computers!"

The Spectrum comes with 16K of RAM on board the basic unit, expandable up to 48K. The keyboard is vaguely reminiscent of the early 8K PETs, but with the layout so designed that touch typing is possible the keys have a nice responsive feel to them, rather than the original 'thump sensitive' keys. As was the case with earlier models each key has a number of functions available on it, including the now famous single key stroke entry for BASIC keywords.

The price of the basic unit is a remarkable £125. This will surely eat its way into the market being slowly established by the VIC and others, and since expansion to 48K costs only £60 more you can start with the 16K version and end up with a 48K machine for only £185. There is a 48K version of the Spectrum available at a cost of £175 cheaper than a 3.5K VIC!

Features present on this machine that were not available on the ZX80 or ZX81 include full colour, sound and high resolution (a resolution of 256 × 192 is available to the user, at no extra cost). These three features will be examined in more detail later.

Availability of the machine is immediate. Sinclair is already geared up to producing 20,000 a month, but even this will probably not meet the anticipated huge demand. Also available



Top, shows use of PAPER command, below two 100K Microdrives, out later

immediately is a printer, the original ZX printer that has been used for some time with the ZX81 is fully compatible.

There is a high degree of compatibility between the ZX81 and the Spectrum. Certainly users of one should have no difficulty in getting rapidly acquainted with the other. The only real problem is the 16K RAM pack for the ZX81, which should not be used under any circumstances, and software cassettes are untransferrable.

Other products announced at the conference are likely to cause an equally large upheaval in the computing industry when they emerge in the coming months. First of these was an RS232/network interface board, costing just £25, and coming out sometime in the autumn of this year. This will enable the Spectrum to be connected up to a whole host of printers, terminals and other computers. The operating systems for this are already designed into the ROM.

Clive diversed from the Spectrum briefly to announce a host of new software for the ZX81, with delivery commencing in May of this year.

Perhaps the biggest rabbit that Clive pulled out of his magician's hat was the ZX Microdrive. This is a very tiny disk drive, using two quarter-inch diskettes, with each diskette capable of holding up to 100K bytes, and a

Review



The key spacing is just enough to allow for touch typing, although our reviewer found it hard going and some people might find the red on grey hard to read

transfer rate of 16K bytes per second. You will be able to connect up to eight of these drives to the ZX Spectrum. Availability was announced as being 'later this year', so anticipate the first arrivals coming sometime in the late summer. The price: £50.

Ordinary cassette decks can also be used as an alternative storage medium.

Ordinary screen resolution of 32 × 24 is adequate for most business (and game!) uses, although this can be enlarged under software control to normal teletext standard.

At a total outlay of just £285 you have a 48K machine with full colour, sound and high resolution, a disk drive giving you 100K of storage, and a printer. The RS232 interface for an extra £20 offers further possibilities.

The hardware

The Spectrum is a smart, slimline machine in black plastic with grey keys, coloured legends and a bright flash of spectral colour across the bottom right corner. Width is 233mm, depth 144mm and height 30mm. Inside the case is a compact printed circuit board containing a handful of discrete components and just 14 integrated circuits, the UHF modulator and a pizoelectric speaker. Of the 14 integrated circuits, eight are RAM memory chips.

The others include the 3.5MHz Z80A microprocessor, the 16K byte ROM (containing Basic and operating system software and the character generator) and lastly the special ULA chip made by Ferranti which allows all the complex colour display circuitry to be crammed into a single IC. Expansion sockets are included on the board allowing one to add an extra 32K of user memory on an internally mounted memory expansion board.

The Spectrum keyboard is a great advance on that of the ZX81. It is larger and the moving rubber keytops have a very pleasant touch and a light click feel on contact.

It is possible to make the key action more positive by giving it an audio feedback. This is done with a simple POKE — POKE 23609,255 — this produces a small beep every time a key is pressed. The keyboard has a standard typewriter layout and the keys are sufficiently widely spaced and large enough to allow touch typing, though the reviewer found this only just possible. There are 40 keys in total with both upper and lower case and a capitals lock feature. Like the ZX81 all the BASIC words are obtainable by pressing single keys, this is further enhanced by the addition of the graphics characters, the user definable by pressing single keys.

Like the ZX81 the Spectrum display has 22 lines of 32 characters but here the similarity ends. Each pixel is memory mapped in a high resolution 256 × 192 display. This uses a special 6144 byte display file located at the beginning of RAM memory, each bit of which maps directly into a pixel on the screen.

There are eight foreground and eight background colours, the data on which pair is chosen for a particular character cell being stored in the colour attribute RAM area. The colour file memory area is 768 bytes long and lies directly above the display file. To use the high resolution there are four new commands in Basic, they are:

PLOT x,y — plots a pixel at co-ordinates x,y on the 256 × 192 point display area. A point can be removed by PLOTOVER (the same as UNPLOT on the ZX81).

DRAW x,y — draws a line from the last high-res point plotted using DRAW, PLOT or CIRCLE to the co-ordinates x,y on the screen. There is also a special form of DRAW which instead of drawing straight lines will draw curved lines and arcs. This has the syntax DRAW x,y,a. If a is positive the line curves left and if negative right, a is in radians so if a = pi then a circle is drawn but if a = pi/2 then only a quarter circle is drawn.

CIRCLE x,y,a — draws a circle of radius a with centre at co-ordinates x,y.

Then there is the function POINT x,y which returns the pixel status at the specified co-ordinates.

These commands seem to be well thought out, easy to use and quite fast. The quality of the generated display is comparable the best currently available.

The circle command is excellent and produces a true circle properly adjusted for screen elongation — this is more then most other manufacturers can do. The draw command while very fast and obviously using a good algorithm is a bit difficult to use since the parameters are offsets of the previously plotted point rather than true screen co-ordinates.

This is definitely a colour machine and the quality of the colour display is excellent. There are eight colours, they are black — blue — red —

continued on page 18

Open Forum

OPEN FORUM is for you to publish your programs and ideas. It is important that you make sure your programs are bug-free before you send them in. We cannot test all of them.

As far as possible always send us a computer printed listing, made immediately after you have run the program.

Each program should include a description of what the program does, how to use it and how the program has been written.

Your contributions should be sent to Open Forum, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2H 7HF with a stamped addressed envelope.

We will pay £10 for each contribution published.

3-D Graphics *ZX81*

The following program will print high resolution graphics on your printer. By experimenting with line 120, you will be able to print a number of shapes based on the 'oval viewed from an angle'.

Enter line 1 as described, only enter spaces where space is written.

1 REM U INVERSE COLON RND U INVERSE ? RND INVERSE N SPACE TAN

Before loading the program enter as a DIRECT command,

POKE 16389 124 NEW

where / equals new line.

Line 2 tests to see if you have remembered to reserve some space in memory by doing this.

Because the program is calculating the position of up to 65000 odd locations it will take a long time — as much as 15 to 30 minutes depending on the formula. The best time to run this is when you're getting ready for bed or about to have a meal, but it is fun!

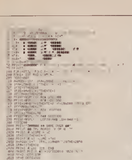
Ikebukuro *Vic*

You are in control of a little man running along the bottom of the screen, who is dodging a rain of arrows coming down from above. These get faster as the game progresses.

To protect you there are three bases which diminish as the arrows wear them down. These can be replenished by getting to the right of the screen, picking up a bag of money and getting back alive: you gain points for each money bag you successfully

3-D Graphics
by Barry Cornhill

Magic Squares
by David Lawrence



Ikebukuro
by Dave Middleton

```
5006 IFTE=1THEN5200
5100 FORI=1TO10 PRINTY$SPC (X)R1$
5105 FORJ=1TO30 NEXT
5110 PRINTY$SPC(X)R2$
5120 FORJ=1TO30 NEXT NEXT GOTO5300
5200 IFC=1?THENXO=16
5202 FORI=1TO10 PRINTY$SPC (X)R3$
5205 FORJ=1TO30 NEXT
5210 PRINTY$SPC(X)R4$
5220 FORJ=1TO30 NEXT NEXT
5300 PRINTY$SPC XY R5$
5310 FORI=1TO2000 NEXT
5315 PRINTY$SPC X R6$ [illegible] GOSUB8000
5320 NI=NI-1 IFNI=0THEN [illegible] GOTO2000
5330 GOSUB6000 GOTO70
6000 PRINT"[illegible]SCORE"[illegible] SC TAB 15/STP[illegible] " [illegible]"
6010 RETURN
7000 SC=SC+INT(RND 0)*10 #100 [illegible] GOSUB3000
7010 PRINT"[illegible]"
7015 CA=CA+2 IFCA+[illegible]THEN[illegible] [illegible]
7020 PRINTLEFT$[illegible]
7030 RETURN
8000 PRINTY$"[illegible]"SPC [illegible]
9000 RN=RND 0)*50[illegible]
9002 IFRN<1THEN9070
9005 FORI=1TORN
9010 PRINT"[illegible]"SPC INT RND 0 *2[illegible]
9020 NEXT
9030 SYS4000+2500
9100 IFPEEK(VP+VD*19+X)=42THEN5000
9110 IFPEEK VP+VD*19+X+1 =42THEN5000
9120 IFPEEK VP+VD*19+X+2)=42THEN5000
9999 GOTO210
20000 DATA 169 0 133 0 169 32 133 1 160 0 169 0 133
20010 DATA 2 169 152 133 3 177 0 201 42 208 57 169 32
20020 DATA 145 0 24 165 0 105 22 133 5 165 1 105 0
20030 DATA 133 6 1[illegible] 5 201 32 208 15 201 160 240 30 201
20040 DATA [illegible] 240 26 109 30 145 5 76 81 26 169 42 145
20050 DATA 5 165 5 133 7 24 165 6 105 120 133 3 165
20060 DATA 4 145 2 [illegible] 200 [illegible] 1 [illegible] 3 [illegible] 189
20070 DATA 1 201 2[illegible] 208 170 56 170 170 170 170 170 170 170
20080 DATA 170 170 170 170 170 170 170 170 170 170 170 170
30000 RESTORE FORI=6656TO6752 READA POKEI A NEXT RETURN
READY
```

carry back. You have three lives in which to amass as high a score as possible.

Magic squares — ZX81

The mathematical oddity of the magic square, where rows and columns of numbers all add up to the same total, has long fascinated mathematicians and masses alike. Using this program ZX81 owners will be able to generate 4×4 magic squares with totals up 3966, in other words up to the point where the square requires more than three digits in one or more of its cells.

These, it should be stressed, are true magic squares, having no duplicate cells. They are also true magic squares in that rather than merely providing the desired totals when rows or columns are added, or even diagonals, the squares generated by this program provide the desired total in no less than 30 different combinations, eg the four corners, the four values in each corner, the four values in the centre.

The program itself simply generates the basic 4×4 magic square with values from 0 to 15 and having a total of 30 and then adds to that square in such a manner as to achieve the desired total.

The values of the square itself are contained in the array M$ which is dimensioned in direct mode, together with N$ which contains the order in which cells within the square are added to in order to achieve the desired value.

The procedure to load the program is as follows.

1. Declare M$(16,3), N$(16,3) and O$(16,3).

2. Enter the temporary input program shown in Figure 2.

3. Enter the following values into N$ 4,7,9,14,1,6,12,15,2,5,11,16,3,8,10,13.

4. Change N$ to O$ in the input section.

5. Enter these values into O$ 9 7,0, 14,6 8,15,1,13,3 4,10,2,12,11,5.

6. Enter the main program and delete the input section, rememberng to start the program with GOTO 1 in order not to wipe out the strings.

7. When requested, enter the number which you wish the rows, columns etc of the square to add up to.

Here are some notes on how the program works.

Line 10. The graphics symbol is that obtained from the '1' key in graphics mode.

Line 20. Graphics symbol on key '4'.

Line 30. This loop copies O$ into M$, thus keeping an uncorrupted copy of the basic square in O$.

Line 90. B is the difference in value between the basic square already stored in memory (30) and the desired square.

Line 100. A useful memory saving trick. This line originally had a line in front of it reading IF B<Y (ie B<4) THEN GOTO 140. CODE "(" is 16 and the form of the line as it now is means that the loop is from 1 to 16 when B>=4 and from 1 to 0 (ie passed by) when B<4. The purpose of the loop is to add 1 to every cell in the square for every whole 4 in B.

Line 140. This loop increments cells to cater for what is left of B when whole 4s have been dealt with by the last loop.

Line 190. The graphics symbol here is that obtained on key '7'. This double loop simply prints out the square.

Because there is nothing to be seen while the program performs its calculations, the program may as well be run in FAST mode.

Read and Data — *ZX81*

The lack of READ, DATA and RESTORE commands on the ZX81 can be very annoying, so machine code program mimics them

Data can be string or numerical, and should be placed in REM statements, each line commencing in a comma, and with each item of data being separated by a comma. For example,

```
100 DATA 4.22,HARRY
```

would be replaced by

```
100 REM 4.22,HARRY
```

There may be as many items of data in each line as you like and lines of data can appear anywhere Ordinary REM statements can be used if the first character is not a comma

Data can be read using the machine code routine starting at 16514 which reads the next article of data and puts it into A$, and returns the length of the data Line 4 in the program makes the variable READ equal to 16514 so USR READ can be used instead of USR 16514

Here are some examples of READ statements

```
90 READ D$
```

would be replaced by

```
LET D$=A$(TO USR READ)
```

and

```
70 READ N
```

would be replaced by

```
70 LET N=VAL A$(TO USR READ)
```

The machine code routine starting at 16601 can be used to set the data pointer to any part in the program The line number to which the data pointer is set is contained in a RAND statement For example,

```
130 RESTORE
```

would be replaced by

```
130 RAND 1
131 RAND USR RESTORE
```

and

```
170 RESTORE 200
```

would be replaced by

```
170 RAND 200
171 RAND USR RESTORE
```

To enter the program type the program as shown, and RUN it from line 10 Type in each number in the object code table When you have finished, lines 10 to 40 may then be deleted, and the program can be saved on tape

The number of characters to which A$ is dimensioned in line 2 is the longest any item of data may be You may alter this if necessary Do not alter line 1, or lines 3 to 6

The program uses memory locations 16507 and 16508 to hold the data pointer Any accompanying Basic program must not use the variable A$ except for in READ statements The machine code subroutines are in the REM statement in line 1

Read and Data
by Simon Lowe

Object code table

```
 42  123   54  120  254   26   40   38
 35   62  117   56  189   46   66   38
 35   36   30  130   43   43  258  234
 46    7   66   36   66   27   39   24
232   38   26   36  125   43   43   43
254   26   30  236   36   35   26  237
 61   18   64   15   16   19   19   15
 19    1    6    6   36   62   36  188
 46   11  199   92  196   66    6  129
 16   16    3   24  239   34  123   84
261   43    1  255  255   24  245   33
  1    6  237   91   54   54   34   58
 64   33  124   64   35  139   16   79
186   46    6   46   11   39   78   36
 78    9   64  266  126  197   56  246
 43   43   34  123   84  261
```

Scrabble scorer — *ZX81*

The program is divided into four sections — picking the first player setting up the display keeping the score and displaying a record

Following the initialisation in lines 10 to 160 which will be explained later the program asks for the names of the players There may of course be two, three or four players and counter P is used to record the number

The maximum name that can be accommodated is eight letters, so Henrietta will have to accept an abbreviation Array T is set up to hold the players' cumulative scores, while S holds the individual scores

The program provides the option of selecting the first player by the traditional method of turning up letters or letting the computer choose In the latter case a pointer moves round the list of names gradually slowing down until it comes to rest

In order to get this name to the top of the list, the names of the players are copied across into array Q$, starting with the first player

The next stage starting at line 600 displays the players' names as they are seated round the table, with the first player's name at the top

We come not to the actual scoring The program always indentifies the next player The score may be input in a variety of ways The most straightforward is of course a single figure, such as 24 but an expression such as 12*2 would be equally acceptance

There is no need to tot up the score, however, because the values of the individual letters are held in array W which was set up by lines 120 to 180 If a word is input, the appropriate score will be added

Double and treble word scores are provided for by lines 1370 and 1430, which will multiply a word score by 2 or 3 if that figure is typed immediately after the word Lines 1281 to 1289, which may be omitted provide a timer

Array X, set up by lines 30 to 110 contains the co-ordinates defining the positions at which the scores will be printed, which again allow for two players sitting 'north' and south

Note the spaces at the ends of lines 1520 and 1530 A nasty pitfall in ZX81 Basic is that if a short number is printed in the position previously occupied by a longer one, the final digits of the longer number are not cleared For instance if a score of 107 were entered, and then corrected by typing -12', the result without the spaces would be not 95 but 957

Finally one of the players may want to check on the score in an earlier round This is provided for by line 1300 If a score of 999 is input, this will cause a jump to line 4000, which prints out array S in the format of Figure 5 This would normally be used at the end of the game so it is terminated by a STOP Should you wish to continue you would need to make a note of the players cumulative scores and enter them again after restarting with RUN

Scrabble Scorer

by Tim Goldingham

```
1 REM SCRABBLE SCORER
2 REM COPYRIGHT T GOLDINGHAM 1982
10 FAST
20 DIM P$(4,9)
21 REM P$=NAMES OF PLAYERS
30 DIM X(4,2)
31 REM X-LOCATIONS TO PRINT TO TALS
40 LET X(1,1)=2
50 LET X(1,2)=15
60 LET X(2,1)=10
70 LET X(2,2)=24
80 LET X(3,1)=17
90 LET X(3,2)=15
100 LET X(4,1)=10
110 LET X(4,2)=4
120 DIM W(26)
121 REM W,Y$=VALUES OF LETTERS
130 LET Y$="13321424185131134111144844"
140 FOR J=1 TO 26
150 LET W(J)=CODE Y$(J)-28
160 NEXT J
170 SLOW
200 CLS
210 PRINT "WHOS PLAYING?"
220 FOR P=1 TO 4
221 REM P=NO. OF PLAYERS
230 INPUT Z$
240 IF Z$="" THEN GOTO 310
250 IF LEN Z$<9 THEN GOTO 280
260 PRINT AT 15,0;"NAME MUST NOT EXCEED 8 LETTERS"
270 GOTO 230
280 LET P$(P)=Z$
290 PRINT AT 5+2*P,5;P;"  ";Z$
300 NEXT P
310 LET P=P-1
320 IF P<2 THEN GOTO 200
330 DIM T(P)
331 REM T=CUMULATIVE SCORES
340 DIM S(50,P)
341 REM S=INDIVIDUAL SCORES
350 PRINT AT 18,0;"TYPE NUMBER OF FIRST  PLAYER,
    OR  ""R""  TO  SELECT  AT  RANDOM"
360 INPUT Z$
370 IF Z$="R" THEN GOTO 430
375 IF Z$<>"R" AND (CODE Z$>37 OR CODE Z$<29) THEN GOTO 250
380 LET N=VAL Z$
381 REM N=NO. OF 1ST PLAYER
390 IF N>P THEN GOTO 250
400 CLS
410 PRINT AT 10,4;"FIRST TO PLAY IS ";P$(N)
420 GOTO 560
430 PRINT AT 0,0;"FIRST TO PLAY IS"
440 PRINT AT 18,0;"
450 LET I=1
460 LET N=INT (RND*P)+1
470 FOR J=1 TO 3
480 FOR K=1 TO P
490 PRINT AT 5+2*K,4;">"
500 IF I=2*P+N THEN GOTO 560
510 PAUSE 3*I
520 PRINT AT 5+2*K,4;" "
530 LET I=I+1
540 NEXT K
550 NEXT J
560 PAUSE 200
570 CLS
600 DIM Q$(P,9)
610 REM Q$=NAMES OF PLAYERS  1ST PLAYER 1ST
620 LET C=1
621 REM C=COUNT OF ELEMENTS OF Q$
630 FOR K=N TO P
640 LET Q$(C)=P$(K)
650 LET C=C+1
660 NEXT K
670 FOR K=1 TO P-C+1
680 LET Q$(C)=P$(K)
690 LET C=C+1
700 NEXT K
800 FOR J=0 TO 25
810 PLOT 19+J,11
820 PLOT 19+J,36
830 PLOT 19,11+J
840 PLOT 44,11+J
850 NEXT J
860 FOR J=4 TO 15
870 PRINT AT J,10;" [illegible]
871 REM (12 * GRAPHICS SHIF[illegible]
880 NEXT J
900 PRINT AT 4,12;Q$(1,1)
910 FOR J=2 TO 9
920 IF Q$(1,J)=" " THEN GOTO [illegible]
930 PRINT Q$(1,J);
940 NEXT J
950 IF P=2 THEN PRINT AT 15,1[illegible]
960 FOR J=1 TO 9
970 IF P=2 THEN FOR J=2 TO 9
980 IF Q$(2,J)=" " THEN GOTO 1010
990 IF P>2 THEN PRINT AT 6+J,23;Q$[illegible]
1000 IF P>2 THEN PRINT Q$(2,J);
1010 NEXT J
1020 IF P=2 THEN GOTO 1200
1030 PRINT AT 15,12;Q$(3,1)
1040 FOR J=2 TO 9
1050 IF Q$(3,J)=" " THEN GOTO 1080
1060 PRINT Q$(3,J);
1070 NEXT J
1080 IF P=3 THEN GOTO 1200
1090 FOR J=1 TO 9
1100 IF Q$(4,J)=" " THEN GOTO 1200
1100 PRINT AT 6+J,10;Q$(4,J)
1120 NEXT J
1200 LET A=1
1201 REM A=ROUND NO
1210 FOR N=1 TO P
1220 LET D=1
1220 REM D=COUNT OF PLAYERS
1230 PRINT AT 20,0;
1240 IF Q$(N,D)=" " THEN GOTO 1280
1250 PRINT Q$(N,D);
1260 LET D=D+1
1270 GOTO 1240
1280 PRINT "S SCORE?"
1281 REM TIMER (OPTIONAL)
1282 PRINT AT 20,24;"        [illegible]
1283 LET T=10
1284 PAUSE 476
1285 IF INKEY$<>"" THEN GOTO 1290
1286 PRINT AT 20,24;J;" SECS $
1287 LET T=T+10
1288 GOTO 1284
1289 REM END OF TIMER
1290 INPUT Z$
1300 IF Z$="999" THEN GOTO 4000
1305 IF Z$="" THEN GOTO 1290
1310 IF CODE Z$>37 THEN GOTO 1340
1320 LET Z=VAL Z$
1330 GOTO 1500
1340 LET SUM=0
1341 REM SUM=VALUE OF WORD
1350 FOR J=1 TO LEN Z$
1360 LET X=CODE Z$(J)
1370 IF X=30 OR X=31 THEN GOTO 1430
1375 IF X>31 AND X<38 THEN GOTO 1290
1380 IF X>31 AND X<38 THEN GOTO 1290
1390 LET SUM=SUM+W(X-37)
1400 NEXT J
1410 LET Z=SUM
1420 GOTO 1500
```

Open Forum

Morra ***ZX81***

This absurd little game from America consists of two players each sticking one, two or three fingers out while at the same time guessing how many fingers their opponents will extend

Guess right and your score is increased by the number of fingers on show from both players, guess wrong and you get nothing The idea is to make a fool of yourself 12 times in this fashion and then declare a winner

The accompanying program plays a mean game of Morra on a 1K ZX81, being based on a mathematical analysis of the game, no less On each turn the computer invites you to guess how many bytes it will display (it could hardly be fingers, could it) and enter how many fingers you will show Then — and it doesn t cheat — the computer declares its guess and the number of bytes

That s about all there is to it, except to remind you that it is definitely cheating to analyse the ZX81 s tactics when you are feeding in the program As if you would

Here are some program notes

Line 100 CODE " " looks a great deal more wasteful than simply 0 which is all it represents That is an illusion based on the fact that we are looking at a display of the program rather than at the program itself In the real program, CODE takes only one byte whereas 0 takes 7 bytes altogether Literal numbers are inordinately expensive in 1K of memory — this program contains no literal numbers whatsoever and would not run if you changed all the CODE values to literal numbers, since there would be too little memory

Line 120. This string and the subsequent loop set the frequency of the different moves and then shuffle the order of the moves

Line 130 Graphics symbol is key '1' in graphics mode

Line 190. Graphics 1 This main loop requests moves and responds 12 times

Line 260. Graphics '4

Line 280. Graphics '4

Line 340. Graphics 'S' Note how logical conditions are attached to the print statements to provide three alternative outputs from one line

Morra
by David Lawrence

```
100 LET [illegible]CODE
110 LET H=C
120 LET H$="11111[illegible]
130 FOR I=CODE " " TO LEN H$
140 LET P=INT (RND*LEN H$+CODE [illegible]
150 LET T$=H$(I)
160 LET H$(I)=H$(P)
170 LET H$(P)=T$
180 NEXT I
190 FOR I=CODE " " TO CODE "[illegible]"
200 CLS
210 PRINT "[illegible] GUESS "
220 INPUT G
230 PRINT "[illegible] OF FINGERS"
240 INPUT F
250 PRINT F "****************"
260 PRINT "MY GUESS " [illegible] CODE " " - VAL H$(I) [illegible]
    VAL H$(I)
270 IF G=VAL H$(I) THEN LET H=H+F+VAL H$(I)
280 IF CODE " " [illegible] VAL H$(I) -F THEN LET [illegible] +F+VAL H$(I)
290 PRINT "****************"
300 PRINT "[illegible] COMPUTER [illegible] HUMAN [illegible]"
310 PRINT AT [illegible] NEWLINE
320 INPUT [illegible]
330 NEXT I
340 PRINT AT CODE [illegible] AND [illegible]
    [illegible] "NO-ONE " AND [illegible]
```

Twenty-one ***BBC***

In this program, you and the BBC Microcomputer take it in turns to throw a six-sided die as many times as you like, trying to get a total of 21, or close to 21, without exceeding 21 ('bust-ing')

This is a dice version of Blackjack, and the computer plays very well You go first in every game entering 1 to roll the die again, or 2 to stand, that is, to stay with the total of 21

There are five rounds to a game and the winner of course is the player who wins the most out of the five rounds

Line 20 sets the mode, and line 30 sends action to a procedure called 'initialise This short procedure (lines 70 to 100) sets the two variables which hold the scores — SI and SM — to zero

From there, line 40 sends the computer s attention to the procedure called 'game_count'. Note that procedure names can be more than one word in length, so long as the underline rule joins the words The 'game_count procedure (lines 110 to 150) set up a loop which calls the procedure 'game five times

You can see from lines 120 and 140 that the BBC Microcomputer supports loop names which are whole words in this case the word, as you can see, is 'count' The procedure 'game which is called from within the procedure game_count, runs from lines 160 to 330

Line 190 is a delay loop, setting the variable T to the value held by the computer s internal clock, and then passing through a REPEAT/UNTIL loop until one second (TIME increments by one every fiftieth of a second) has passed

Line 220 waits for the player's input, and line 230 rolls the die, and adds its value to the score (line 200) if the player has decided to keep rolling

If not, the computer checks (line 240) to see if the player has exceeded 21 (and, if so, goes to the end_of_game' procedure) If the player has not busted, the computer acknowledges the total the human has decided to stick with (line 250) and proceeds to roll the die itself

A three-second delay (line 280) comes before the computer announces the value it has rolled, and its total Lines 300 to 320 check the computer's score, and compares it with the player's score, and decides whether or not to roll again

The 'end_of_game procedure (lines 340 to 380) looks at the two final totals and decides which player has won After five rounds have been played (counted, you'll recall, by the loop in the 'game_count' procedure), action goes to the finale procedure (lines 390 to 470) to find the winner of the five games

Open Forum

Twenty-one

by Tim Hartnell

```
[illegible] TWENTY ONE
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible] ROC
[illegible] PROCgame_[illegible]
[illegible]
[illegible]
[illegible] NEXT count
[illegible] ROC
[illegible] PROCgame
[illegible] =0
[illegible] update
[illegible] TIME=0:REPEAT UNTIL TIME > 50
[illegible] PRINTCHR$(129);"TOTAL [illegible] ;E
[illegible] PRINTCHR$(131); 1 TO ROLL, [illegible] TO STAND"
[illegible]
[illegible] E=E+RND(6): GOTO190
[illegible] PRINT CHR$(129);"YOU'VE BUSTED!!"; PROCend_of_game:ENDPROC
250 PRINT CHR$(133);"OK, YOU STAND ON ";E
[illegible]
[illegible]
[illegible] REPEAT UNTIL TIME > 150
[illegible] "I rolled a [illegible] so my total is ";F
[illegible] PROCend_of_game:ENDPROC
[illegible] PROCend_of_game:ENDPROC
[illegible] AND E 22 THEN 260
[illegible]
[illegible] of_game
[illegible] PRINT'CHR$(1.9);"DEAD HEAT":SI=SI+1:SM=SM+1
340 [illegible] E AND F 22) OR E 21 PRINT CHR$(129);"I WIN":SI=SI+1
370 IF [illegible] F 21 OR(E F AND E 2) PRINT'CHR$(133); YOU WIN":SM=SM+1
380 END[illegible]
390 DEF [illegible]ale
400 [illegible] (130);"FINAL SCORES.'
410 [illegible] (129); YOU: ";SM,"ME: ";SI'
420 T=TIM[illegible]EAT UNTIL TIME [illegible]
430 IF [illegible] PRINT'I ";
440 IF [illegible] PRINT'YOU ";
450 IF SI [illegible] PRINT WE BOTH ";
460 PRINT [illegible] "WIN "
470 END[illegible]
480 DE[illegible]
490 [illegible] ££££££££££££££££££££££££££££££"'
500 T=TIME:REPEAT UNTIL TIME [illegible]
510 PRINT [illegible] ;count
520 PRINT [illegible] YOU ;SM, [illegible] ;SI
530 ENDPROC
```

Review

continued from page 11

purple — green — blue — yellow — white

Any character location can have two colours, a foreground colour — this is the colour of the dots which make up a character — and a background colour In Spectrum Basic the foreground colour is known as INK and the background as PAPER, the colour of either being specified by following the INK or PAPER command with a number from 1 to 6 which specifies the colour, or attributes as they are referred to in the manual

The command BORDER will set the colour of the border around the screen display area The command INVERSE can be used to display characters in reverse field — foreground colour becomes background colour and vice versa OVER is an interesting command since it allows a character to be printed over the existing contents of that character square without obliterating it

The brightness of the display in a particular character space may be changed by either the commands FLASH or BRIGHT FLASH as its name implies causes the specified characters to flash between a very bright screen and a normal screen BRIGHT enhances the specified characters by making them brighter than normal There is also a colour function ATTR (line, column) which return the colours, both foreground and background of the specified character

The Spectrum incorporates an inbuilt loudspeaker which can produce a sound with a frequency range of over 10 octaves The sound can also be output to an amplifier via a jack socket at the rear To generate a sound there is a special command in Basic called BEEP

The BEEP command is followed by two variables The first determines the frequency of the note and the second the duration The quality of the produced sound is not marvellous but quite adequate for adding nice sound effects to games programs

The Spectrum is broadly software compatible with the ZX61 ZX81 Basic is a subset of Spectrum Basic except for the following features The FAST/SLOW commands have been abolished The SCROLL command is unnecessary since Spectrum automatically scrolls UNPLOT has been replaced by PLOTOVER The character codes used are the standard Ascii characters rather than the special ZX81 codes

The memory map of the Spectrum is quite different to that of the ZX61 and the following are some of the salient features

0 — 16383	ROM operating system and kernal
16364 — 32767	Internal 16K of RAM
32768 — 65535	expansion RAM area
16384 — 22528	Display file 6144 bytes
22529 — 23296	colour attribute file
23297 — 23552	printer buffer
23553 — 23734	system variables
23735 —	microdrive maps, channel information etc then Basic program storage area

The cassette interface has been entirely changed from the ZX81 and tapes recorded on the ZX61 will not load on the Spectrum The reason is that the tape data transfer rate has been increased very considerably — up to 1500 baud Verification and error trapping techniques have also been improved

The RS232 network interface board will be very useful since besides allowing the user to construct local area networks with Spectrum computers and peripherals it will also enable the Spectrum to be connected to any of the myriad different computers and peripherals with RS232 interfaces The potential here is enormous large-scale networks via cheap modems, Spectrums as low-cost terminals to larger microcomputers or even mainframes

The Microdrive disk drive is a sensation — the Japanese have been talking about it but Sinclair has done it Capable of storing 100K on a 2 25 inch disk with a data transfer rate of 16K bytes per second — the average access time is less than 3 5 seconds, and all for a price tag of about £50 It will be available in the early autumn when we will give you a full review The Basic commands required for the disk drive are included on the Spectrum so there is no need to upgrade your machine The commands are OPEN and CLOSE, MOVE, ERASE, CAT and FORMAT

The documentation for the Spectrum is good and perfectly adequate for the average user Sinclair has learnt a lot from the ZX61 manual which has been applied to the manuals There are two for the Spectrum There is a simple small 30-page introductory manual and a large 220-page main programmer s reference manual

Conclusion

This new computer from Sinclair clearly represents excellent value for money and will, no doubt, prove a great success

The basic 16K version of the Spectrum costs £125 There is a 48K version at £175 Availability is immediate and Sinclair is geared up to produce 20,000 per month

Sinclair also announced that later this year he would release a 100K ZX Microdrive using 2¼ inch disc, costing £50 Up to eight of these can be attached to the Spectrum The Microdrive will be reviewed when it becomes available

The RS232 interface costs £20

It's time to tune up your Vic to concert pitch

Last week I programmed the Commodore Vic-20 computer to play 'user friendly tunes' This week I will make the tunes a little more user-friendly

I don't really like the phrase 'user-friendly', but what I will do is make the program a little easier to use, especially if your main interest is in making music using established musical principles

Before I describe how the program has been updated, I ought to warn the musical purist about the music creation facilities on the Vic-20

Anyone who has an ear for music might notice that the notes generated by the Vic are not exactly pure, or, more to the point, are only approximations

The nearer you reach the upper limits of the Vic's musical range, the more approximate these notes become. However, there is no need to be too pedantic

The reason for this 'loss-of-tune' — if that is the right phrase — is that the Poke numbers given by Commodore for the Vic synthesiser are universal The same Pokes that give the note of C in, say Japan, are also used to invoke C in the islands off the coast of Scotland This would be fine, if all the Vics used the same clock speeds Unfortunately because the world's power supplies are not all the same, the clocking speed of the world's Vics are not all the same

```
50 REM MUSIC LEVEL 2
60 PRINT"[CLR] INPUT NOTES AS LETTERS   TOPOCTAVE NOTES
     PRECEDED BY ↑"
100 REM
200 REM  INITIALIZE
300 REM
400 V1=36874   V2=36875   V3=36876   V4=36677
     V5=36878
500 REM  SET VOLUME
600 POKE V5,7
700 REM
800 REM INPUT NOTES
900 REM  KEY OF C
1000 REM
1100 DIM A(14),B$(14),S$(3)
1200 FOP I = 1 TO 14  READ A(I)   READ B$(I)
     NEXT I
1300 DATA 195,C,201,D,207,E,209 F,215,G,219,A,
     223 B,225 ↑C,228,↑D,231 ↑E 232,↑F 236,↑G
1350 DATA 237,↑A,239,↑B
1400 REM
1500 REM INPUT USER DEFINED TUNE IN LETTER FORM
1600 REM
1700 PPINT PRINT" HOW MANY NOTES "   INPUT N
1800 DIM NN(N,3)
1900 PRINT"[CLR,RVS] ENTER TUNE[OFF]" PRINT" VOICE1
     VOICE2,VOICE3,[RVS]RETURN[OFF]"
2000 FOR I=1 TO N  FOR T=1TO3 INPUT S$(T)  NEXT T
2100 FOR T=1 TO 3   FOR J= 1 TO 14  IF S$(T)=B$(J)
     THEN NN(I,T)=A(J)
2200 NEXT J NEXT T
2300 PRINT"[CD]"
2900 NEXT I
3200 REM
3300 REM NOW PLAY TUNE
3400 REM
3500 FOR I=1 TO N
3600 POKE V1,NN(I,1) POKE V2 NN(I,2) POKE V3
     NN(I,3)
3700 FOR J=1 TO 100   NEXT J
3800 POKE V1,0 POKE V2 0   POKE V3,0
3900 NEXT I
4000 PRINT "PLAY TUNE AGAIN"  INPUT W$   IF W$="Y" THEN
      GOTO 3500
```

Getting the drift of a melody

That is why melodies played on the Vic often appear to drift in and out of tune But we should be happy to live with these imperfections, after all, it is such imperfections that make most musical instruments sound so individual Soon clever people will be standing in bars, listening to the juke boxes and saying 'Hey listen to that funky Vic bass line!'

Last week's music program was fine — that is it worked OK, and it also produced some fairly interesting music Would you believe I even managed to play some Bach on my Vic? It made a welcome change from all that Dépêche Mode stuff which is beginning to drive the neighbours up the wall

There is however one minor problem with the program described last week — to enter notes you first have to look that note up in a table I thought it would be a darn sight better to simply enter the name of the required note straight away, and let the computer do the converting After all that's what it's supposed to be good at

Two problems arise in making the program easier to use — neither of them insurmountable First, to use more than one octave — look the word up if you don't understand — there must be some way of telling the computer which octave is in use I chose to precede a letter by the ↑ (up arrow) symbol to indicate the higher octave Second there is a little problem with the inputting routine

In this week's program the line numbers have been changed a bit, I'm sorry about that, but it shouldn't be too confusing One change that might catch you out is in lines 1100 and 1200 — note that DIM sizes and FOR values are now 14, rather than 15 The DATA statement in line 1350 is new and line 1300 has been extended to list the names of the notes Other changes are in the inputting routines, which are now more complicated, and the output routine which is now simpler All these extra changes have been made at a cost of about eight notes off the maximum length of tune possible **Sam Blythe**

Up-to-date information is the key to control

***David Kingsbury** explains the importance of the process status register*

The 6502 has a primitive set of jump instructions when compared to the Z80 microprocessor

One important thing that you must understand in any microprocessor that you wish to program in assembly language is the processor status register (PSR) The PSR gives you information about the instruction that you executed most recently

The 6502 PSR has seven bits that you can get at to control the microprocessor Five of these bits reflect the outcome of the last operation and two (the Interrupt and Decimal flags) are used to force the 6502 to behave in a particular way

The diagram shows the PSR The example program using the BEQ instruction looked at the Zero bit of the PSR and altered the course of the program according to whether or not there was a 1 or a 0 in bit 1

Now suppose you are writing a word processing program in which you need to increment (add one) or decrement (subtract one) from a counter that keeps track of how many letters you have typed You will need to use a two byte counter because that can count up to 65535 where a single byte (8 bit) counter can only count up to 255

You may need more than one counter in the program, for example to keep track of the cursor when you are editing a document, to record the current end of text position

If you place the counter bytes next to each other in Zero page memory then you need only one subroutine to increment any counter Here it is

```
          REFERENCE - 80 (Memory location in
                             Page Zero RAM)
          TXEND   TCURS   PCR   ECHO
LOW BYTE  80      82      84    86
HIGH BYTE 81      83      85    87

          PHP
          TXA
          PHA
          INC REFERENCE X
          BNE JIM
          INX
          INC REFERENCE X
JIM       PLA
          TAX
          RTS
```

And to decrement one of the counters you could use this subroutine

```
          PHP
          PHA
          TXA
          PHA
          SEC
          LDA REFERENCE X
          SBC £1
          STA REFERENCE, X
          BCS JAN
          INX
          DEC REFERENCE, X
JAN       PLA
          TAX
          PLA
          PLP
          RTS
```

If you load Register X with zero it will increment or decrement the counter labelled TXEND while if you load X with four before entering the subroutines PCR will be affected



Programming

Increase the power of your Sinclair machine

***David Lawrence** explains how to simulate the powerful DEF FN function on a ZX81.*

Non-ZX81 owners who own one or other of the more expensive micros often comment on the Sinclair machine's lack of the DEF FN function

This is a powerful facility, available in many Basics, which allows the user to define a function once in the program, and from then on, rather than spell it out every time, the user can simply refer to it by a shorthand name

For instance DEF FNA (X)=10*X sets up a simple function which multiplies a variable by 10. On subsequent occasions in the program any variable can have the same operation performed on it, for example, LET Z=FNA (Y) would result in Z being set to 10 times Y

Multiplying a number by 10 may not be a very impressive achievement but the function defined could just as easily have been a formula 20 terms long, which could have resulted in considerable memory saving and ease of programming

Even so, ZX81 owners should not despair, since the Sinclair micro's extraordinarily flexible VAL function can be made to perform most of the functions of DEF FN and some that exceed most other machines. The tiny program shown is a pointer to how this can be achieved

The program is intentionally tiny and in its present shape is of limited use because it shares the memory of an unexpanded 1K ZX81 with two string matrixes — A$ (10 20) and B$ (10 8) — leaving not a great deal of room for unnecessary frills

You can use long formulae

But the program is capable of processing data according to any one of 10 formulae up to 20 characters long, with up to 8 variables in each and with all the necessary variables being defined by the use of only one input line

Needless to say, to design a 1K program with 10 lines of complex formulae, together with input lines for up to 8 different variables for each, would not be possible. It is achieved here by storing the formulae as strings in A$

Here, for example, are 5 formulae as stored in the array

```
X*((1+R)**-N)
R*D/100
N*N*(N+1)**2/4
2*V*V*SIN T*COS T/9.81
SQR (U*U+2*F*S)
```

Parallel to these are the lists of necessary variables stored in B$ 1-5

```
XRN
D
N
VT
UFS
```

A powerful calculator

The functions of the formulae are unimportant, they are shown purely for illustration. The important point is that, by Poking the variable name location in the input statement at line 100, the single loop at line 50 can read the appropriate line in B$, obtain the necessary variable names and have them defined in what is, when lines which are not strictly necessary are removed, a space of 5 lines

This would be true whether there were 10 formulae or 100 on an expanded ZX81 and, having obtained the values of the variables the formula can be assessed in a single line by the VAL function, as at line 140

In the case of the particular program shown, the result is that a 1K ZX81 is turned into a powerful calculator for those who work with the same set of formulae a great deal, but have not previously been able to fit them into the unexpanded machine

To set the program up, the formulae you wish to use must be entered in direct mode, eg, LET A$(1)="X*(1+R)**-N", together with the list of variables required, in this case all that would be needed is LET B$(1)="XRN". Once the formulae are entered, and there can be up to 10 using the 10,20 dimensions mentioned earlier for A$ (more if your particular formulae are shorter) all the user has to do is to specify a number and the particular formula will be displayed, together with a series of requests to define the variables

The program, however, is only one example of the power of the VAL function. Once stored in a string, your formulae can be called up at any time during the execution of the program. All that is necessary is that the variables be already defined then a simple LET A=VAL A$(1) will execute whatever is stored in the first line of A$, a real space saver if the same operation is to be performed several times

The major limitation, as compared with the DEF FN of other Basics is that a formula can't be defined in terms of 10*X, as in our first example, then made to work on Y instead but that limitation is easily overcome by simply operating on X and then setting Y=X, a small sacrifice in return for the ability to define far more than the 26 functions most Basics provide for

Program notes

Line 30. It is not usual to use a 2 character variable in a 1K program that is short on space but here it is important to have a variable name that will never be duplicated in any of the formulae

Line 50. The 2 CODE values are simply to cut down on the number of literal numbers and, hence, on memory demands

Line 70. If the character referred to in B$ is A, line 100 becomes INPUT A

Line 80. The same operation is performed on line 110

Line 140. Hey presto! the result of the formula is displayed

Line 150. This input is purely a device to leave the formula, variable values and answer on display until you press NEWLINE

MICROTANIC SOFTWARE

ADD A PRINTER TO YOUR 6502 SYSTEM FOR LESS THAN

£90

Connect a

SINCLAIR ZX PRINTER

(£59.95)

to your 6502 COMPUTER with:

PRINTERFACE 1 for Microtan
PRINTERFACE 2 for Micron/Tan rack
PRINTERFACE 3 for ATOM/Acorn rack
PRINTERFACE 4 for AIM-65/KIM/SYM
PRINTERFACE 5 for BBC COMPUTER
PRINTERFACE 6 for VIC 20

Please ask for others

PRINTS UP TO 43 CHARACTERS/LINE
OR 256-PIXEL GRAPHICS
PRINTERFACE UNIT AND 600-byte
PROGRAM LISTING: £29.95 inc VAT

NOW AVAILABLE FOR
MICROTAN 65
ACORN ATOM
BBC COMPUTER
AIM 65/KIM/SYM
VIC 20
MICROTAN SYSTEM RACK

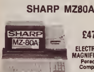
SEND CHEQUE/POSTAL ORDER PLUS 50p POST AND PACKING TO

MICROTANIC SOFTWARE
235 FRIERN ROAD, DULWICH, LONDON SE22
or tel 01-693 7659
PRICES INCLUDE VAT

Croydon Micros

★★★ IN STOCK ★★★

SHARP MZ80A



£477

ELECTRONIC MAGNIFICENCE
Personal Computer

Over £50-worth of FREE PROGRAMS including BASIC tutorial, 10 games, home finance, and GEMINI tape database

01-689 4349

★★ Now in our new premises ★★
20-28 Whitehorse Road
Croydon CR9 2NA

ALSO RC1500 Pocket Computer — £145

Prices exclude VAT   FRONTREALM LTD (T/A)

VIDEO SOFTWARE LTD

NEW VIDEO-INDEX NEW

Q How do you store 57,000 characters in a 16K RAM?
A. Use VIDEO-INDEX

Catalogue your cassettes, your record collection, references to magazine articles. Amazingly versatile. Capacity for 1 000 individual references each of which can generate 57 characters of text. Powerful search facilities once the index has been created.

How is it done?

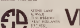
The system cannot really fit 57 000 characters in a 16K RAM. It gives the illusion of doing so by eliminating duplicates. For instance in an index of articles in computer magazines about the ZX81 certain words occur time and again. VIDEO-INDEX detects these duplications and thereby conserves space.

What do you get?

1. A machine code master program which sits at the top of RAM. This is initially loaded like a BASIC program.
2. A demonstration index containing 1 000 references to articles about the ZX81 in the popular magazines.
3. A detailed instruction manual.

You may then proceed to create your own catalogues and indexes and save them on cassette. This program is fast, efficient and ingenious and is by far the best product we have introduced for the ZX81.

N.B. There isn't room for a commentary on the cassette so you'll have to read the manual. ZX81 or ZX80+8K ROM. ZX printer is useful but not essential. Price £9.95



VIDEO
SOFTWARE LIMITED

Peek & poke

Peek your technical queries to Ian Beardsmore. He will poke back an answer

BUT WILL IT MAKE YOUR EYES HURT?

Ron Cryer of Grants Avenue Bournemouth writes

Q I have a BBC Microcomputer model A, and want to be able to use different coloured backgrounds and text, but find the explanations given for use of the VDU command far too complicated for me to understand Will you tell me, in simple words exactly how I can get, for example, a green background with red words

A Just enter and run the following routine, and you'll soon learn how the different numbers produce different effects Entering some higher numbers will produce some most peculiar results, as I'm sure you'll discover Keep a notebook handy, and write down the combinations of A and B which produce the effects you most like

```
10 REM **Determining colours**
20 REM **Model A BBC Micro**
30 MODE4
40 PRINT"
50 INPUT"BACKGROUND NUMBER
   A
60 INPUT"FOREGROUND NUMBER
   "B
70 VDU 19,3,B,0,0,0
80 VDU 19,0,A,0,0,0
90 GOTO 40
```

TURN THAT CURSED THING OFF

H McWilliams of Conningham Road, London W12 writes

Q I have been working with an Acorn Atom at school for some months now, trying to write interactive programs I've managed some fairly primitive Space Invader' ones, and a Shoot the Target game, but I find that although the computer runs quickly enough in Basic to produce a worthwhile game, the flickering cursor (which is very big in the standard text mode) is a nuisance I can't find out how to turn the cursor off I would be very grateful if you could explain this for me

A It is fairly simple to turn the cursor off All you need is a simple Poke command Just enter at the start of your program ?hashE1=0 and the cursor will disappear Pressing BREAK then OLD will get it back as will ?hashE1=128 You can in fact, make the cursor just about anything you want Try Pokeing this address with other numbers and see what you come up with

THERE WILL NOW BE A SHORT DE . . . LAY

Matthew Buckre of Hyde Heath Amersham writes

Q I've noticed that some Basics like the one available on the BBC Micro computer, allow for an INKEY$ with a specific built-in delay, so it waits for say half a second for a key entry, before continuing Is there any way of providing a delayed INKEY$' like this on the ZX81, as this would be very useful for me in a program I am writing

A A simple routine should enable you to emulate the delayed INKEY$ to some extent You'll have to experiment to find out the most suitable value to check for in line 30 (where there is an 8 in the example)

```
 5 LET S=1
10 LET A$=INKEY$$
20 LET S=S+1
30 IF S<8 AND INKEY$=  THEN
   GOTO 10
```

DE FINEST FUNCTION YOU CAN FIND

Charles Vitrios of Roby Merseyside writes

Q I have a Sinclair ZX81 and I've noticed many programs in the computer magazines which I would like to run on my computer However they are not written in Sinclair Basic and in many cases I don't know how to convert them

As I am doing a computer science course at school many of the programs I'm interested in are mathematical One thing I come up against time and again is DEF FN and FN How can I define and call functions on a ZX81? And what about the other commands when I don't even know what they're supposed to do? How can I convert them?

A DEF FN to define a function is available in many versions of Basic, but not on the ZX81 In some cases, assigning strings, and then using VAL can help you out of a spot, but this does not give you anything like a true FN call which allows you to specify the argument of the function The best way around the problem is to write out the function in full each time it is called in the program

On the more general problem of converting from one Basic to another there is a book which can be of great help It is called *The Basic Handbook (An Encyclopedia of the Basic Computer Language)* and is written by David Lien Looking up an obscure command or statement in this book will give you many equivalents for it, and once you know what the Basic statement is meant to do, will make it fairly easy to work out an equivalent in ZX81 Basic

YOU TOO CAN HAVE A SUPER POWER . . .

Sally Francis of Woodlands Avenue, London W3 writes

Q I have only 1K on my ZX81 and I am very frustrated by the tiny amount of memory available Could you tell me some ways of saving memory, so I can get more program in? It is particularly difficult to edit lines when the memory is nearly full Is there some way of doing it easily

A There are a number of useful ideas for saving memory on the ZX81 For example you can set a variable equal to the number one at the beginning of the program and use it whenever you need to use one (like at the start of a FOR/NEXT loop) You can say LET A=PI/PI or LET A=SGN PI to set A to one It is cheaper', in terms of the memory you use up, to then say LET B=A+A than it is to say LET B=2

The VAL function is very useful LET A=VAL A$ where A$="13245" is much more memory-efficient than saving LET A=13245 The codes of characters can also be used if you want fairly low numbers For example LET A=CODE inverse space" will set A=128, while saving you a few bytes in the process

To edit on a ZX81 when the memory is nearly full type in LIST n where n is the line number you want to edit Then press CLEAR, and when the screen goes clear press the EDIT button and the line you want will appear at the bottom of the screen where it can be edited

ER, EXCUSE ME IF I'M A BIT BASIC

Peter Overton of Little Stoke Road, Stoke Bishop writes

Q You may think my questions are a little naive, but I find myself quite bamboozled when I read computer magazines I do not yet have a computer, although I've been promised a ZX81 for my birthday and have been buying various magazines to try and learn a bit about them Could you tell me, in plain simple language, the difference between ROM and RAM?

A Don't worry about asking what other people may think of as simple questions We all have to start somewhere, and the arcane language used by many computer users is not as difficult to master as it may appear to be the first time you hear it ROM stands for Read Only Memory and RAM for Random Access Memory The ROM part of the computer is all the things it knows permanently, things it does not forget when the power is turned off like how to add numbers together or how to produce a picture on the TV screen By contrast RAM is the temporary memory storage in which you can write your own programs The computer forgets everything in RAM when the power is turned off

Send your questions to Peek & poke *Popular Computing Weekly* Hobhouse Court 19 Whitcomb Street London WC2 7HF

Classified

Do you want to sell your computer and buy a bigger and better one?

Have you ever thought of trying to make some money out of selling tapes of your own programs?

Whatever it is you want to buy or sell why not use our classified pages? It has to be better than waiting for up to nine weeks to get into one of the old monthly magazines

Not only that, but our rates are very reasonable

For private individuals it only costs 20p per word, with a minimum of 10 words. We can make it so cheap because we charge companies using the classified columns 40p per word

The classified pages can be used for semi-display advertising. The cost for this is £10 per single column centimetre, with a minimum charge of £30

All copy for the classified pages must be pre-paid Cheques/postal orders should be made out to *Popular Computing Weekly*, and your advertisement should arrive at least two weeks before the publication date.

For semi-display advertising please call David Lake on 01-839 2846

For classified entries please fill in the form below and send it, with your cheque/postal order, to Classified Department, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2

If you have any queries regarding classified advertising please call **01-839 1855.**

Please write your copy in bold capital letters on the lines below.

Here's my classified ad.

I make this .... words at .... per word so I owe you £

Name ....

Address ....

....

....

Telephone ....

Please cut out and send this form to
Classified Department, *Popular Computing Weekly* Hobhouse Court 19 Whitcomb Street London WC2

Competitions

1 Here's how you can win Vic-20 Super Expander

Each week we plan to give away a valuable piece of hardware as a prize for the best program written by you, the readers on a theme set by us

The program must be no longer than 100 lines and be suitable for the Vic 20, the ZX81 or the BBC Micro

Your entry should be accompanied by an explanation of the function and use of the program and also of how you went about writing it

The winner will be the reader who in the opinion of the Editor of *Popular Computing Weekly* has sent in the most original and inventive program.

In all cases his decision is final

From time to time we will also publish programs which did not win the competition out which we think will be interesting to other readers

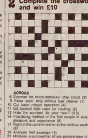
If that happens we will pay the usual rate for contributions to the magazine

This week's fantastic prize is a ZX81 16K RAM pack and the theme for the competition is **Moon Landing**.

Send your program to The Editor *Popular Computing Weekly* 19 Whitcomb Street London WC2

Please mark your entry Hardware Competition 2 — and remember it must reach us by first post May 10 1982

2 Complete the crossword and win £10



ACROSS

8 Summer for micro-computer chip circuit (5)
9 False peer king without lock cleaner (7)
10 Cut basic integer operation (4)
11 US football field used for cooking (8)
13 Hide the sun-bed lie your hear? (3 3)
14 Interfacing method of the first couple in duty pleasure and experience (6)
17 Right in the current neither to the north or south (6)
19 Analysis half passage (4)
21 Mixtures a pul together all are genuine beer (4 3)
22 Mini tulle in France (5)
23 The edge of insolence (3)

DOWN

1 Court call as upsetting as SAs (11)
2 Not even love notes a strange rhyme (3,3)
3 Lament for leading cryptic clue (3)
4 Sons of East Russian operators (6)
5 Audio interface bear before drink (7)
6 Support for the second standard (4)
7 Basic iteration during nearest circuit (3 4 4)
12 Not worth considering for no short time a student (7)
15 Stretch basically the number of bytes (6)
16 Jury rigged plane (5)
18 Stone circuit rising under another (4)
20 A representative current (3)

3 Solve the puzzle and win a gift voucher!

On a desert island were five castaways They had collected a pile of coconuts which they agreed to divide into equal shares the following day

That night the first man awoke, and, fearing that he might be cheated out of some of his coconuts decided to take his share whilst the other men were asleep

He went over to the pile of nuts, and divided them into five equal shares There was one coconut left over so he gave it to a passing monkey

After hiding his share he piled the remaining nuts back into one heap and went back to sleep

Then the second man woke up, he too had the same idea Like the first man he too had one nut left over after dividing the pile of nuts into five This he too gave to the monkey After hiding his share he went back to sleep

As the night progressed the remaining three men all carried out exactly the same procedure The next morning they all woke and succeeded in dividing the remaining coconuts equally between them

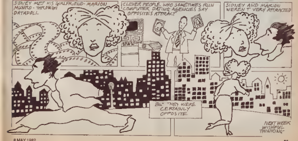
Can you discover (a) What is the smallest number of coconuts that were originally collected? (b) If when making his final division, there was still one left over for the monkey how many would there have been in the original pile?

Closing date for both the crossword and the puzzle is the Monday, three weeks after the cover date

Please mark your envelope 'CROSSWORD or PUZZLE

CITIZEN PAIN

BY DAVID IRELAND and JAMES MACDONALD




BYTESHOP COMPUTERLAND
VIC
VIC-20